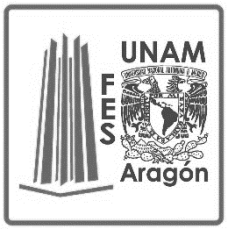

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

Ingeniería en Computación

COMPILADORES

Tarea

Profesor: Marcelo Pérez Medel

Leonardo Olvera Martínez

Grupo: 2608 (2024-II)

Fecha: jueves 29 de febrero de 2024

Tarea A

Al observar el video "El Último Sefaradí", me sumerjo en una época crucial de la historia que fue el año 1492. Este año no solo marcó la culminación de ocho siglos de coexistencia en Al-Andalus, sino también la expulsión de árabes y judíos de España. La narrativa destaca la tragedia de aquel tiempo, donde más de 100,000 sefardíes judíos se vieron obligados a convertirse al cristianismo o enfrentar el exilio.

La parte del video que aborda la diáspora y el legado sefardí me lleva a reflexionar sobre la resistencia y la adaptabilidad de una cultura frente a la adversidad. A pesar de más de 500 años de expulsión, la presencia de aproximadamente 100,000 sefardíes judíos en Jerusalén desafía las predicciones de que su cultura estaba destinada a extinguirse. Es un testimonio de la fuerza de la identidad cultural y la capacidad de las comunidades para mantener su legado a través del tiempo y el espacio.

La parte del video, centrada en la preservación de la identidad y la cultura entre los sefardíes modernos, resalta la importancia de la lengua y las tradiciones en la transmisión intergeneracional de la herencia cultural. La enseñanza del ladino por parte de Eliezer no solo es un acto de preservación cultural, sino también un recordatorio de cómo el lenguaje puede ser un vínculo vital con la historia y la identidad.

Personalmente, este video me ha llevado a reflexionar sobre la importancia de comprender y recordar nuestra historia compartida. ¿Cómo podemos aprender de los errores del pasado para construir sociedades más inclusivas y tolerantes en el presente? La preservación de la identidad cultural, la valorización de la diversidad y la reflexión crítica sobre las decisiones políticas son elementos cruciales para forjar un futuro más comprensivo y respetuoso.

Tarea B

```
"""
Esta funcion sirve para ver los separadores de un texto, nos servirá para
separa en tokens nuestro codigo
"""
def esSeparador(c):
    separadores = "\n\t " #salto de linea, tabulaciones y espacio
    return c in separadores

"""
Esta funcion detectará algun simbolo especial, de esta manera será mas sencillo
asignarle esa etiqueta a los tokens
"""
def esSimboloEsp(c):
    especiales = "¡#$%&/*+-=:;[]{}(),"
    return c in especiales

"""
Esta función es algo extensa, será la primera que usemos y nos sevirá para
quitar los comentarios de nuestro codigo, de esta manera, sin comentarios,
posteriomente podremos empezar a separar en tokens
"""
def quitaComentarios(cad):
    # estados: A, B, C, Z.
    """Segun lo visto en clase, aprovechamos el uso de las maquinas de estado
    para saber en que punto estamos, a continuacion se explicara como funciona"""
    estado ="Z"    #representa el estado inicial
    #cad = "a=b/c;"
    cad2 =""        #creamos una cadena temporal donde almacenaremos el codigo
    for c in cad:
        if (estado=="Z"): #revisamos si estamos en el estado incial
            if (c=="/"): #si encontramos un / puede comenzar un cometario
                estado = "A" #cambiamos a estado A donde PODRIA ser un comentario
            else:
                cad2 = cad2 + c # si no comienza comentario metemos en cad2
        elif (estado=="A"): # estamos en el estado donde PODRIA se un comentario
            if (c=="*"): # sigue un *, estamos seguros que es un comentario
                estado="B" #cambiamos a estado B donde estamos ya en comentario
            else: # si no recibimos un * puede que sea una divicion
                estado = "Z" #volvemos al estado de inicio
                cad2=cad2+"/"+c # agregamos el / que nos saltamos en el tado A
        elif (estado=="B"): #sabemos que estamos en el estado con comentarios
            if (c=="*"): # hasta que no encontremos un * no meteremos nada a cad2
```

```
            estado = "C" #encontramos el * PUEDE que termine el comentario
        elif(estado=="C"): #Estado donde PUEDE terminar elcometario
            if (c=="/"): # si encontramos / ha terminado el comentario
                estado="Z" #volvemos al estado de incio, Z
            else: # de no ser asi el comentario continua
                estado="B" #volvemos al estado B donde sigue siendo un comentario
    return cad2

"""
Esta función servirá para empezar a separar por tokens nuestro codigo ya sin
comentarios
"""
def tokeniza(cad):
    tokens = [] #Creamos un array donde metermos nuestros tokens
    dentro = False #variable que usaremos en caso que no sean un simEspecial
                   #es posible que sea un id, palabra reservada o un tipo
    token = "" #variable donde guardaremos los casos anteriormente mencionados
    for c in cad:
        if dentro:  # es un caso de los anteriores mencionados
            """
            Esta funcion es en caso que no sea un simbolo especial, significa
            que puede ser o un id, palabra reservada o un tipo, sabemos que
            estos tokens acaban cuando hay un espacio, el primer if se encarga
            de ello, cuando encuentre un espacio meteremos nuestra variable
            temporal y meteremos lo que almacenos ahi en el array, de igual
            manera pasará si encontramos un caracter especial, mientras no pase
            eso, todo lo que encontremos irá a la variable temporal
            """
            if esSeparador(c):
                tokens.append(token)
                token = ""
                dentro = False
            elif esSimboloEsp(c):
                tokens.append(token)
                tokens.append(c)
                token = ""
                dentro = False
            else:
                token = token + c
        else: #Iniciamos con este, ya que dentro por defecto es falso
            if esSimboloEsp(c): #si es un simbolo especial lo mete al Array
                tokens.append(c)
            elif esSeparador(c): # si es un separador no hace nada
```

```
            a=0
        else:
            dentro = True # de no ser alguna de las anteriores lo manda
                    # a la funcion para meter el conjuto en un token
            token = c
    return tokens

"""
Esta funcion nos servirá para saber si es un ID
"""
def esId(cad):
    return (cad[0] in "_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ")

"""
Esta funcion nos servirá para saber si es una palabra reservada
"""
def esPalReservada(cad):
    reservadas = ["main","char", "int","float","double","if","else","do",
                  "while","for","switch","short","long","extern", "static",
                  "default","continue","break","register","sizeof","typedef"]
    return cad in reservadas

"""
Esta funcion nos servirá para ver si es un tipo de cadena
"""
def esTipo(cad):
    tipos=["int", "char", "float", "double"]
    return cad in tipos



# Programa para probar las funciones
codigo_prueba = """
/* Este es un comentario */

int main() {
    /* Declaración de variables */
    float var1 = 3.14;
    char char1 = 'a';

    /* Estructuras de control */
    if (var1 > 0) {
```

```
        print("La variable 'var1' es mayor que 0\n");
        return 1;
    } else {
        print("La variable 'var1' no es mayor que 0\n");
        return 0;
    }
}
"""

# Prueba de las funciones
print("Código prueba:")
print(codigo_prueba)

print("Resultado después de quitar comentarios:")
codigo_sin_comentarios = quitaComentarios(codigo_prueba)
print(codigo_sin_comentarios)

print("Tokens resultantes:")
tokens_resultantes = tokeniza(codigo_sin_comentarios)
print(tokens_resultantes)

print("Pruebas de identificadores, palabras reservadas y tipos:")
for token in tokens_resultantes:
    if esPalReservada(token):
        print(token + " es una palabra reservada.")
    elif esTipo(token):
        print(token + " es un tipo de dato.")
    elif esSimboloEsp(token):
        print(token + " es un simbolo especial.")
    else:
        print(token + " es un identificador.")
```

```
Código prueba:

/* Este es un comentario */

int main() {
    /* Declaración de variables */
    float var1 = 3.14;
    char char1 = 'a';

    /* Estructuras de control */
    if (var1 > 0) {
        print("La variable 'var1' es mayor que 0
");
        return 1;
    } else {
        print("La variable 'var1' no es mayor que 0
");
        return 0;
    }
}

Resultado después de quitar comentarios:



int main() {

    float var1 = 3.14;
    char char1 = 'a';


    if (var1 > 0) {
        print("La variable 'var1' es mayor que 0
");
        return 1;
    } else {
        print("La variable 'var1' no es mayor que 0
");
        return 0;
    }
```

Ln: 80 Col: 22

```
['int', 'main', '(', ')', '{', 'float', 'var1', '=', '3.14', ';', 'char', 'char1
', '=', "'a'", ';', 'if', '(', 'var1', '>', '0', ')', '{', 'print', '(', '"La',
'variable', "'var1'", 'es', 'mayor', 'que', '0', '"', ')', ';', 'return', '1', '
;', '}', 'else', '{', 'print', '(', '"La', 'variable', "'var1'", 'no', 'es', 'ma
yor', 'que', '0', '"', ')', ';', 'return', '0', ';', '}', '}']
Pruebas de identificadores, palabras reservadas y tipos:
int es una palabra reservada.
main es una palabra reservada.
( es un simbolo especial.
) es un simbolo especial.
{ es un simbolo especial.
float es una palabra reservada.
var1 es un identificador.
= es un simbolo especial.
3.14 es un identificador.
; es un simbolo especial.
char es una palabra reservada.
char1 es un identificador.
= es un simbolo especial.
'a' es un identificador.
; es un simbolo especial.
if es una palabra reservada.
( es un simbolo especial.
var1 es un identificador.
> es un identificador.
0 es un identificador.
) es un simbolo especial.
{ es un simbolo especial.
print es un identificador.
( es un simbolo especial.
"La es un identificador.
variable es un identificador.
'var1' es un identificador.
es es un identificador.
mayor es un identificador.
que es un identificador.
0 es un identificador.
" es un identificador.
) es un simbolo especial.
; es un simbolo especial.
```